(12) UK Patent Application (19) GB (11) 2 206 428 (13) A

(43) Application published 5 Jan 1989

(21) Application No 8713918

(22) Date of filing 15 Jun 1987

(51) INT CL[4]
G06F 15/31

(52) Domestic classification (Edition J):
G4A FGK

(56) Documents cited
EP A 0235764
EP A 0206580
EP A 0201797
EP A 0132926
WO A 87/01485
US 4247892

(58) Field of search
G4A
Selected US specifications from IPC sub-class G06F

(71) Applicant
Texas Instruments Limited

(Incorporated in United Kingdom)

Manton Lane, Bedford MK41 7PA

(72) Inventor
Simon C J Garth

(74) Agent and/or Address for Service
Abel & Imray
Northumberland House, 303–306 High Holborn, London, WC1V 7LH

(54) Computer

(57) A computer comprising a three-dimensional array of data processing units. The computer has particular application in simulating neural networks in which case the data processing units operate as neuron simulators. Each data processing unit produces the sum of input numbers multiplied by respective coefficients and performs a non-linear operation on the sum to produce an output number. The data processing units have serial data intercommunication links 21–26 between adjacent units along the coordinate axes of the array, the units at the boundary of the array being connected to inputs and outputs. The intercommunication is provided by a shifting register in each unit, the registers of two units being joined end to end in a loop through multiplexers and data in each register being shifted into the other register when data communication between the units is required. A broadcasting means may be provided to enter data and/or instructions into all or selected ones of the data processing units directly.

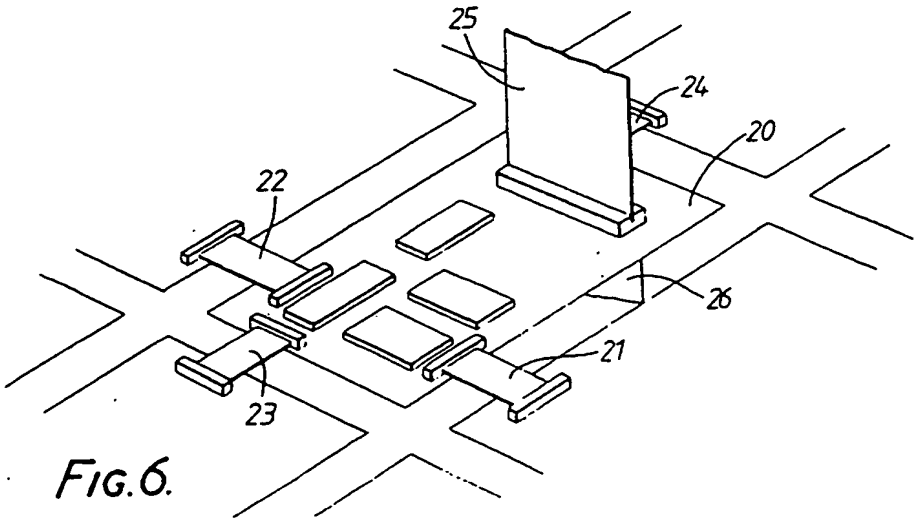

FIG.6.

The drawing(s) originally filed was (were) informal and the print here reproduced is taken from a later filed formal copy.
The claims were filed later than the filing date within the period prescribed by Rule 25(1) of the Patents Rules 1982.

BNSDOCID: <GB 2206428A>

GB 2 206 428 A

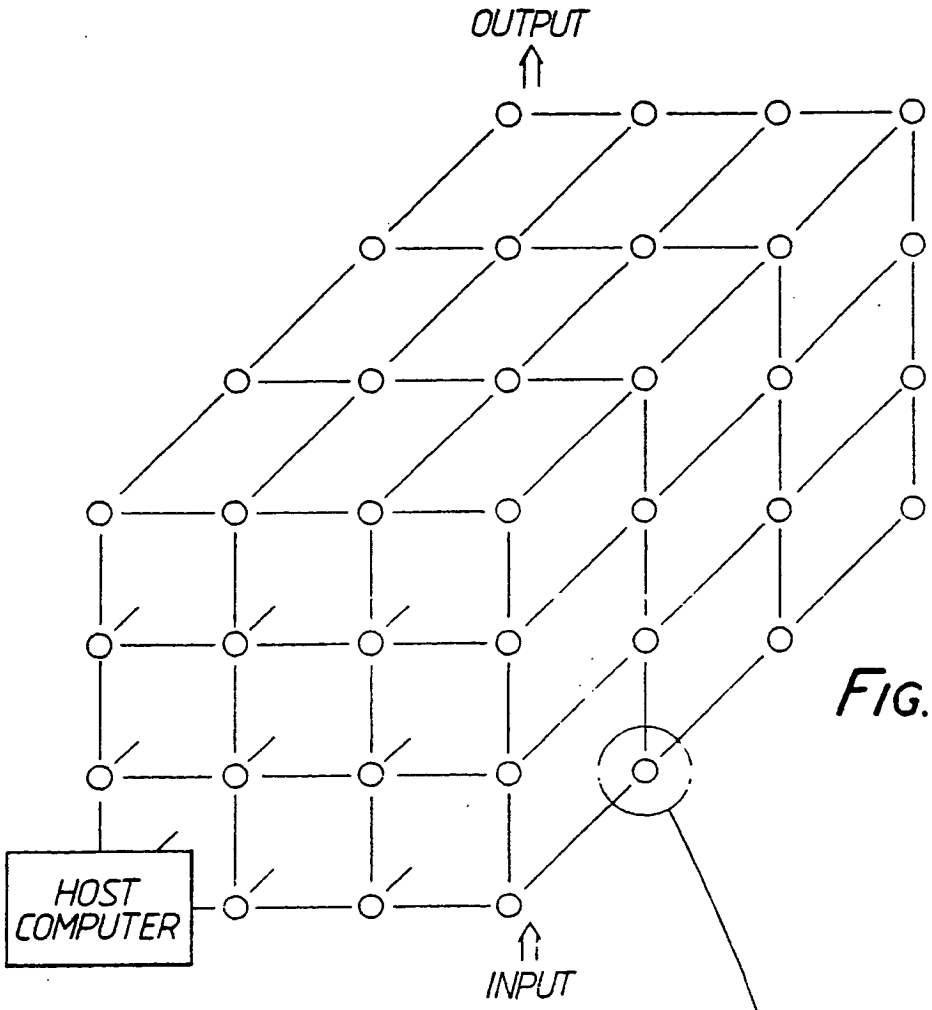

FIG. 1.

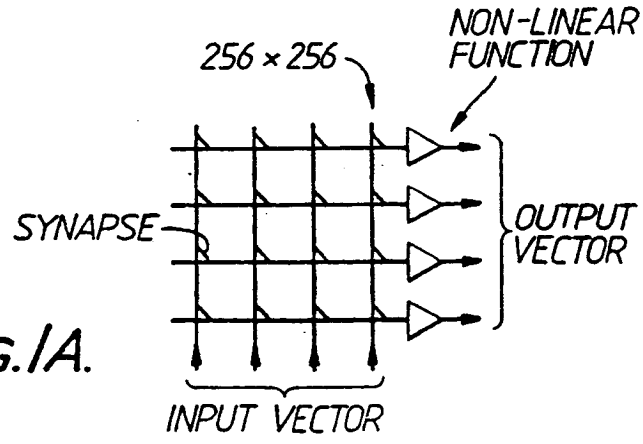

FIG. 1A.

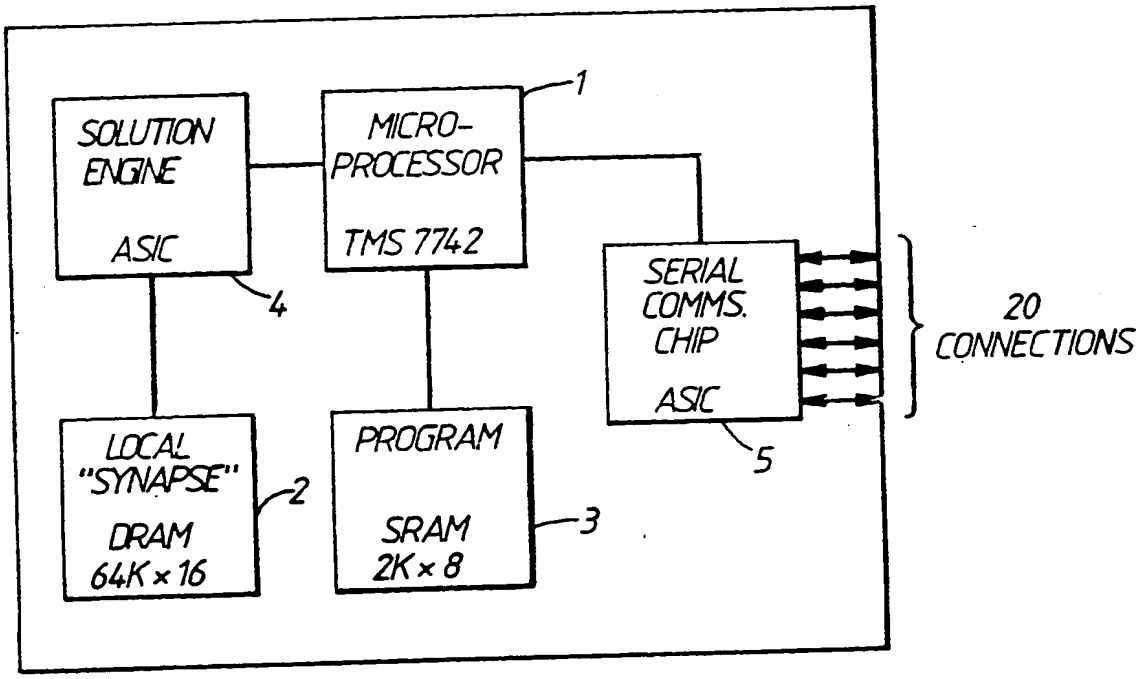

FIG. 2.

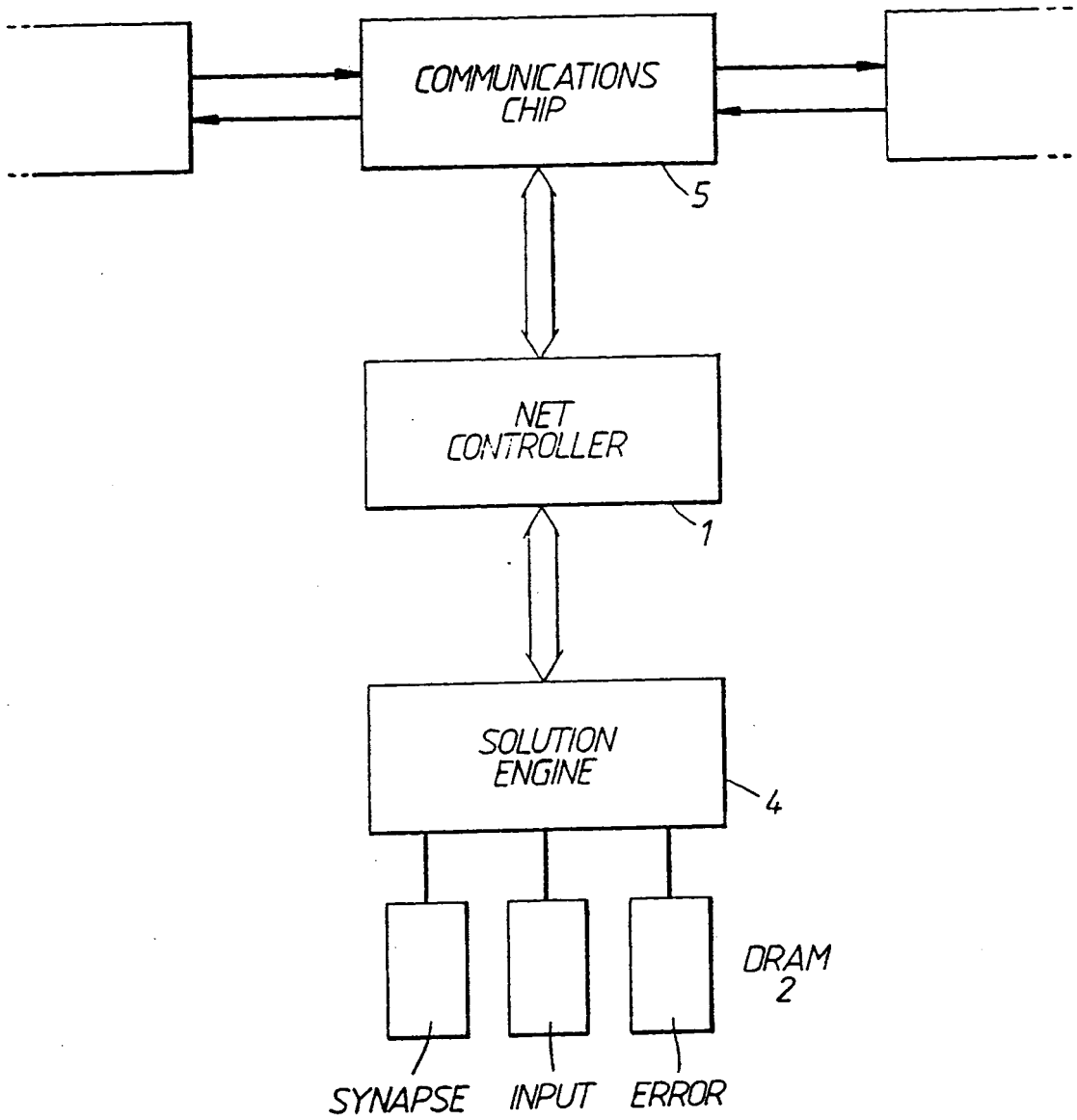

FIG. 3.

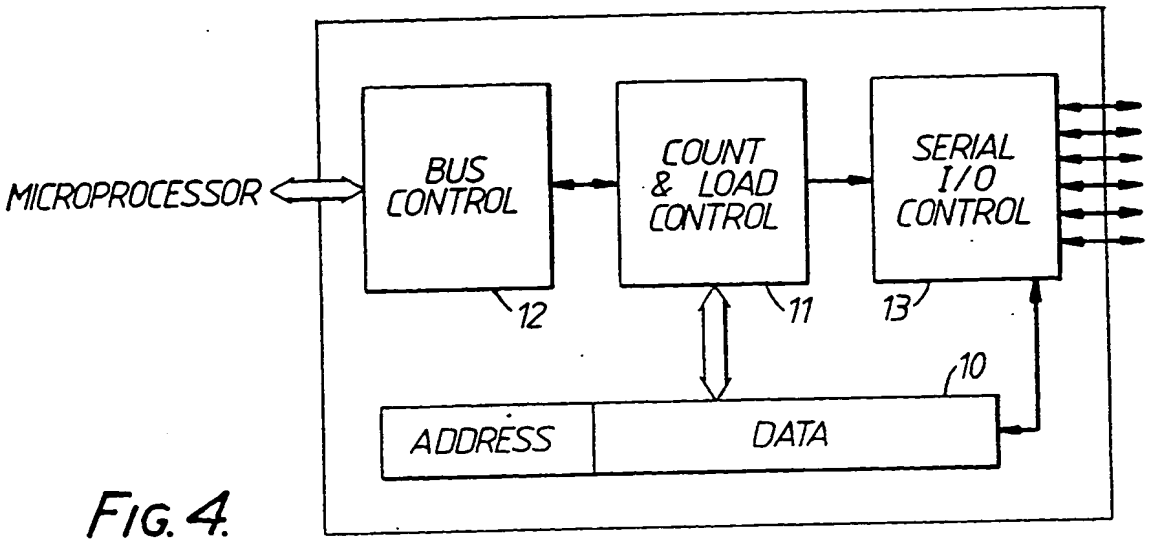

FIG. 4.

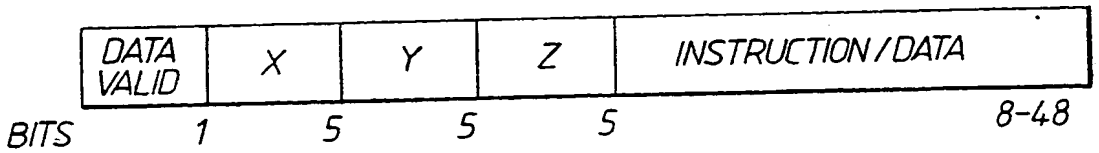

FIG. 5.

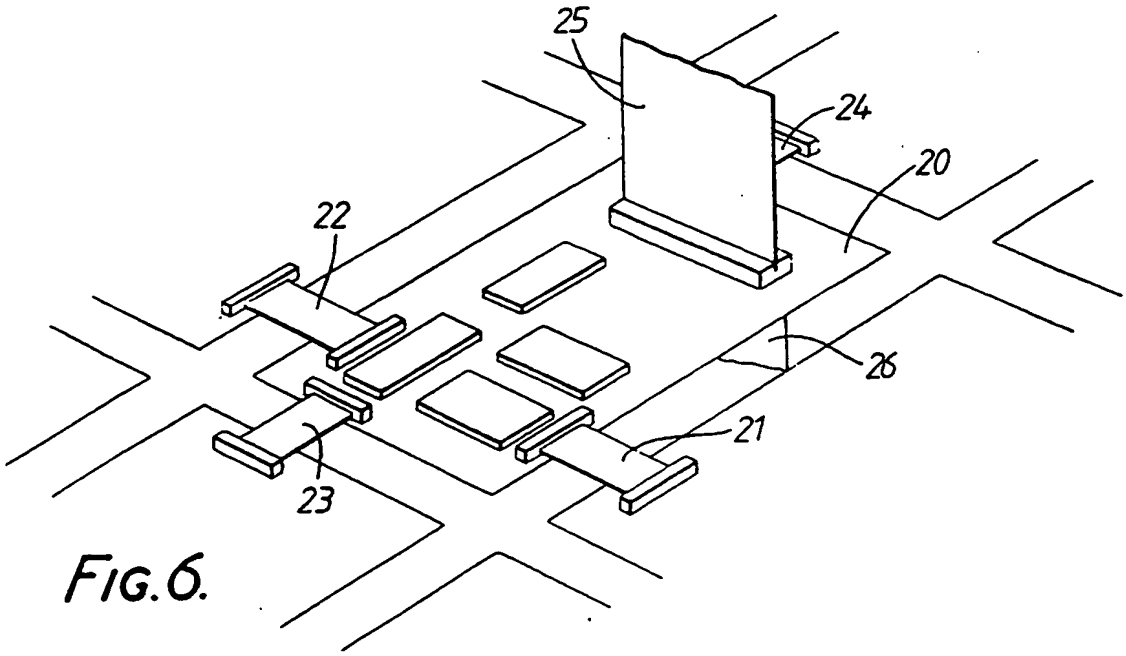

FIG. 6.

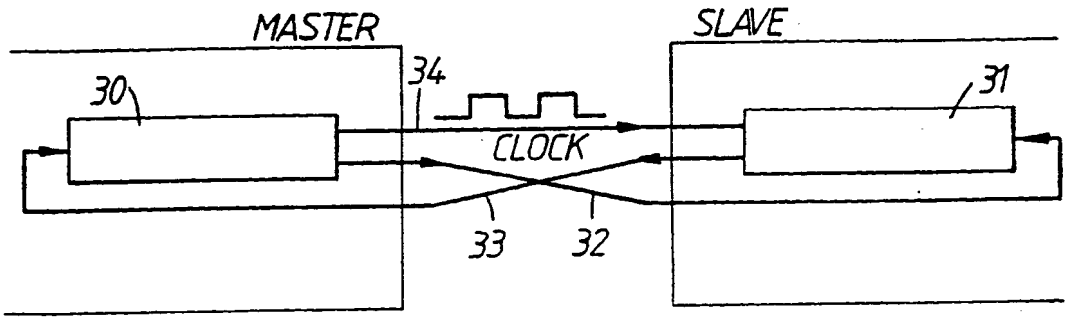

FIG. 7.

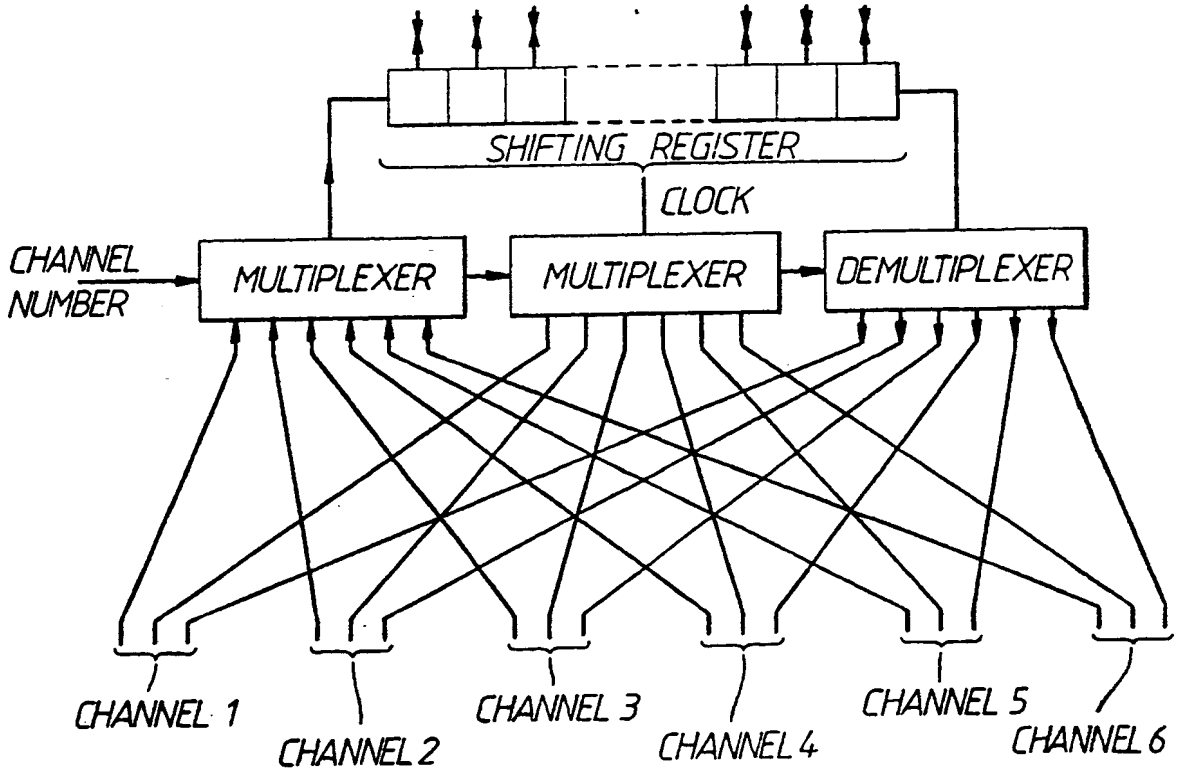

FIG. 8.

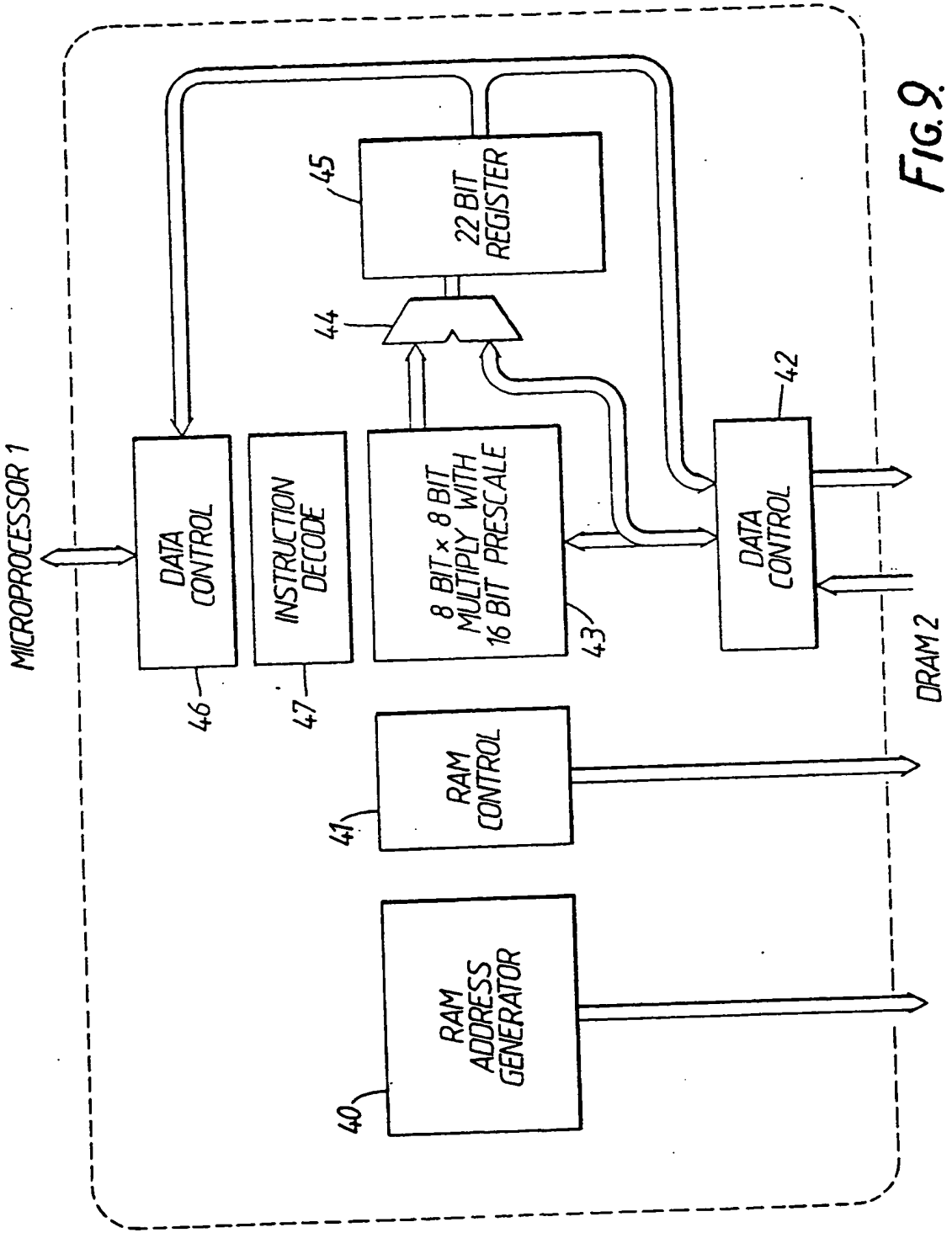

FIG. 9.

COMPUTER

This invention relates to computers and in particular to a highly parallel computer having a plurality of interconnected processing units capable of operating simultaneously.

In particular for recognition tasks it is required to simulate the operation of a large neural network having, for example, 1000 by 1000 interconnected neural units (neurons). A particular feature of such machines which is required to be exploited is their ability to learn to recognise a characteristic of the input signals by adaptive techniques in response to the error between the output of the machine when it receives a particular input and the output which it is required to produce. Past work has involved the study of a large highly interconnected network and more recently it has been shown that many groups of individual networks which are locally highly interconnected can operate satisfactorily with comparatively sparse interconnections between the groups. There has been a problem in setting the memory elements in such systems to give a desired response characteristic, and this problem has been met by the introduction of "back propagation" or "error propagation" which is a technique defining a procedure by which a system with an arbitrary number of groups of neural units having no direct connection with the inputs or outputs of the system can learn from the input data and the error between the actual output and the desired output of the network.

The simulation of a neural network falls into two phases, solution and learning.

In the solution phase the input group of numbers or vector to a unit (neuron) is multiplied by corresponding stored elements (synapses) in that unit and the resulting products are summed to form a single number. The single number is then applied to a non-linear operator the output of which forms the output of the unit and is communicated to the input of one or more other units. From a network of units the outputs of the final units form the output of the network.

The learning phase makes the use of the error propagation algorithm and in this phase an error vector is computed from the difference between the output from the network and the desired output and the stored elements in each unit are updated by the product of the error vector and the input vector giving rise to it.

Both the solution and learning phases involve a large amount of calculation and a lot of accesses to the memory storing the elements. Neural networks have been simulated on large scale computers, but it has been found that the time involved for a single processing unit to perform all the calculations is prohibitively high and for a large network could involve as much as several months of calculation. This is clearly impractical.

It is an object of the present invention to provide a computer specifically for the purpose of simulating neural networks

According to one aspect of the present invention there is provided a computer comprising a plurality of separate data processing units each including a processor, a multi-address read-write memory and communication means, wherein the data processing units are arranged in a three-dimensional, possibly rectangular, array and the communication means of each unit is connected to the communication means of the nearest units only in each direction along the three axes of the array.

According to a second aspect of the present invention there is provided a computer comprising a plurality of separate data processing units each including a processor, a multi-address read-write memory and communication means, the communication means of each unit being connected to the communication means of at least two other data processing units or to the communication means of at least one other data processing unit and to a data input or a data output, wherein each data processing unit operates as a neuron simulator responding to a plurality of data bits as an input and producing a plurality of data bits as an output.

The processor may include a gate array configured as a neural network solution engine capable of multiplying a plurality of input numbers by respective coefficients, summing the products to produce a single number and performing a non-linear operation on the single number to produce an output number.

The data processing unit may include a microprocessor as a controller for the unit.

The communication means may be arranged to operate serially, having a shifting register with, say, 64 stages to which the data to be communicated to another unit is transferred in parallel. Communication between data processing units is effected by connecting the shifting registers of the two units end to end in a ring and clocking the data from each shifting register into the other. The communication means may have only a single shifting register the end terminals of which are connected selectively to pairs of terminals of the data processing unit by multiplexers. A data processing unit may have six pairs of terminals which are connected to pairs of terminals of six other data processing units. Control of the multiplexers, for example by a microprocessor in the unit determines the identity of the other data processing unit to which the shifting register is connected. A link between the clocks may be provided to synchronise the stepping of the data in the two connected shifting registers.

Each data processing unit is preferably connected by its communication means to only the nearest other data processing units along each coordinate direction of the array in which the units are arranged. This simplifies the communication requirements for the units without restricting the ability of each unit to communicate with each other apart from the imposition of a small time delay.

The computer may further include broadcasting means connected to a receive-only input of each data processing unit enabling messages and commands (such as start, reset and stop) to be sent directly to each unit. It can also be used to transmit large amounts of data to the data proces-

sing units, for example, initial synapse values for setting up the computer. Selection of the units capable of receiving certain messages from the broadcasting means may be effected by allocating multi-bit addresses to the units and providing the messages with a multi-bit address code, each unit being arranged to compare that code with its own address and only accepting messages addressed to them. The comparison of address codes may be restricted to say "1" bits only so that a message for all units would have an address code of all 1's.

In order that the invention may be fully understood and readily carried into effect it will now be described with reference to the accompanying drawings, of which:-

FIGURE 1 is a diagram of a three-dimensional array of interconnected simulated neurons;

FIGURE 1a is a simplified diagram of one such simulated neuron illustrating its function by analogue computing circuitry;

FIGURE 2 shows an example of a digital circuit used to form a simulated neuron according to the invention;

FIGURE 3 is a functional diagram of the circuit of Figure 2;

FIGURE 4 is a diagram of the communications chip used in Figure 2;

FIGURE 5 illustrates the format of a message from one neuron to another;

FIGURE 6 is a perspective diagram of a neuron according to Figure 2 showing the interconnection paths to adjacent neurons;

FIGURE 7 is a diagram of part of the communications chip of Figure 4 illustrating the mechanism of serial data transfer;

FIGURE 8 is a diagram to be used explaining the operation of the circuit of Figure 7; and

FIGURE 9 is a diagram of an example of the solution engine shown in Figure 2.

The example of a computer to be described is designed specifically to simulate large numbers of neural networks in

parallel. The overall structure of the machine is shown in Figure 1 and consists of a large number (tens or hundreds) of individual autonomous neural networks. The networks are implemented in groups using digital data processing units represented by circles in Figure 1. A plurality of these units are interconnected to form a three-dimensional array to which input data is applied at one side and from which output data is derived from another side. The entire system is controlled by a host computer located at one corner of the array which issues global commands such as 'start' and 'stop' to the data processing units, is the means by which parameters are transferred to the neural networks (e.g. representing the interconnections for data to be implemented between the networks) and performs the data gathering and error reporting functions of the system. The host computer may be a conventional microprocessor or a more powerful one such as, for example, a mini or main frame computer.

Each data processing unit, of which a simplified diagram is shown in Figure 1A, may be regarded as a configurable neural network consisting of, for example, 256 simulated neurons each with typically 256 inputs. The inputs are represented in Figure 1A by vertical lines and the neurons by horizontal lines. At each cross-point in the network there is located a synapse, a 16-bit integer shown diagrammatically as a triangle in Figure 1A. Solution of the network is performed by applying an input vector, of which each member is an 8-bit term to the inputs, forming the product of the input vector terms and the corresponding synapses is formed for each neuron and summing the products for each neuron to give a 24-bit number. This number is then operated upon with a non-linear function to give an 8-bit element of the output vector for each neuron. The non-linear function may be such as to give an output value tending towards unity as the input value increases to a maximum positive value and towards zero as the input value falls to a maximum negative value, symmetrically disposed about a middle input value where the output value is ½. One such non-linear function is $1/(1+e^{-x})$. In this way for an N x M matrix (N

neurons having M inputs per neuron) an M input vector is transformed into an N element output vector, this process constituting the solution of the network. The output vector from one neural network forms the input vector of one or more networks and the process is continued until the entire system of networks has been solved. In the data processing unit the multiplications by the synapses, the additions and the non-linear functions are implemented digitally.

The learning phase is executed in a similar manner, the precise sequence of calculation steps depending on the algorithm used. In all cases the learning phase results in the updating of the synapse values stored in each data processing unit in dependence upon differences between the actual output of the system and its desired output.

The updatting of the synapse values may be based on the error propagation algorithm proposed by Rummelhard and McCelland in "Parallel Distributed Processing" published by MIT press 1986. The error, equal to the difference between the desired output of the network and the output actually produced, is propagated backwards through the neurons as were used in the solution process. For each neuron a local error is produced on the assumption that the local errors of the neurons connected to drive the same neuron are proportional to the output values of the neurons. Using the local errors the values of the synapses are updated so as to tend to reduce the local errors to zero using

$W_{ij} = W_{ij} + \eta \Delta_i I_j$, where W_{ij} is the synapse value for the j^{th} input which contributed to the local error Δ_i, and η is a scaling factor (< 1). In this way, the adjustment of the synapse values serves to reduce to zero the global error of the whole network.

Figure 2 shows a block diagram of one data processing unit of the type used in Figure 1. It consists of standard components, a microprocessor 1, a dynamic RAM 2 of capacity from 64k x 16 bits upto 4M x 16 bits and a static RAM 3 of capacity 2k x 8 bits and two semi-custom gate array circuits, a solution engine 4 and a serial communications circuit 5. The components are interconnected in conventional

manner by data, address and control buses. The whole unit is mounted on a printed circuit board approximately 3" square and has the ability to solve a 256 x 256 network in 20 ms. The gate array chip 4 forming the solution engine is arranged to perform both the solution and learning operations. The serial communications circuit 5 acts as a general purpose input/output unit for communication along the three dimensions of the array which allows the results of the computations to be transmitted to a destination unit within the time taken by the unit to solve the network so that the communication imposes no delay on the solution process.

The microprocessor 1 acts as the controller for the unit to provide flexibility of operation. Since the multiplications and additions are implemented by the solution engine 4, it is not necessary for the microprocessor to be particularly fast. The TMS 7742 microprocessor indicated in Figure 2 includes 4k bits of EPROM in which is stored the kernel software to perform boot strapping, a self-testing routine, the control signals required for the other components of the unit, error handling and a simple independent network solution routine. The static RAM 3 provides additional program space for use by the microprocessor 1 in which a user can store his own network solution program and algorithms for the learning phase. This program information is transferred from the host computer (Figure 1) using the serial communications system to be described. Obviously, if it were decided to use the microprocessor to perform additional functions it might be advantageous to use a more powerful one in order to maintain the speed of operation of the data processing units.

Figure 3 is a functional diagram of the data processing unit shown in Figure 2, showing the connection of the communications circuit 5 to the controller, that is to say the microprocessor 1, which in turn feeds the solution engine 4 which is connected to the dynamic RAM 2 storing the synapse values, the terms of the input vector and the errors arising during the learning phase.

A typical operation during the solution phase is as

follows.

The microprocessor 1 instructs the solution engine 4 to evaluate one neuron. When the multiplications and addition are complete and the output value obtained the microprocessor 1 receives the result, applies the non-linear function to it, computes the location for the output of that function and transfers the data and the address to the communications circuit 5. While the microprocessor is performing these operations the solution engine 4 has been evaluating the next neuron and the microprocessor then receives the result of that. The successive results derived by the data processing unit are transferred serially between the unit and the appropriate nearest neighbouring unit in the three-dimensional array. This process continues until the final results are transferred to the outputs of the entire system. When a unit receives data via the communication system its microprocessor causes the value to be stored in the appropriate region of the dynamic RAM for use in evaluating the neurons of that unit. It will be appreciated that the non-linear function applied to the sum produced at a neuron and the interconnections between data processing units can be chosen freely since they are defined in the software stored by the microprocessor. The gate array 4 forming the solution engine has no flexibility beyond the variation in the values of the synapses used in the multiplication operations and can therefore be arranged to handle its restricted range of operations at very high speed.

There are many possible ways of interconnecting a plurality of data processing units into an array or a net. Since it is not known at the outset to which other units the outputs of a unit will need to be transferred, it is desirable that the interconnection should allow the transfer of a message from any one unit to any other unit. However, to provide separate connections between each pair of units in a group would result in an enormous number of connections being provided. This difficulty can be overcome by arranging to pass messages from each unit to each of its nearest neighbours from which by successive transfers a message can

be ultimately be passed to the required destination unit.

As indicated in Figure 1, the units are arranged in a three-dimensional array which lends itself to a layered planes architecture with communication between units in either direction along each of the co-ordinate axes.

The communication itself which is effected by the serial communication circuit 5 of a unit involves the production of a message for transmission, the message having a 16-bit address and up to 48 bits for data. The message is stored in a 64-bit shifting register 10 shown in Figure 4 by a count and load control circuit 11 which receives its data from the microprocessor 1 via a bus control circuit 12. When the message is ready for transmission the serial input/output control circuit 13 implements the serial transfer of the data from the register 10 to a corresponding register in the destination unit. A typical data rate for serial transfer is 10 Mbits/s and the communication link is sufficiently fast that the time required to pass a message is likely to be shorter than the computation time for a neuron in all but the most adverse circumstances. The serial communication link is also used for the transfer of input data and instructions from the host computer to the data processing units and for the collection of output data by the host computer.

Figure 5 shows in more detail the constituent parts of a message. The 16 bits of the address portion consist of a single bit indicating that the message is a valid one, and three 5-bit groups representing the X, Y and Z co-ordinate values of the unit in the array to which the message is addressed. The single validity bit is required because of the use of the technique, described later, of exchanging the contents in two registers to effect the transmission of a message from one unit to another. If the data exchanged for a message is not another message the validity bit will indicate this and the unit receiving the data will ignore it.

Figure 6 is a perspective diagram showing a data processing unit 20 with six communication links 21,22,23,24, 25,26, to the nearest adjacent units in the three co-ordinate

directions.

In the operation of the serial communications system, messages are passed by connecting the ends of the transmitting and receiving message registers in the communication circuits of the units concerned so as to form a continuous loop and the contents of the registers are then swapped by shifting the data out of one register serially into the other.

Figure 7 is a diagram showing the principle of the serial communication system where a register 30 in one unit is connected by conductors 32 and 33 to a register 31 in the other unit to form a continuous loop. A third conductor 34 is connected between the registers to ensure the synchronism of the clocks stepping the bits along in the two registers. For such transfer one unit is designated the 'master' and the other the 'slave' so that the clock of the master controls the clock of the slave. This master/slave designation is purely arbitrary, and the unit nearer the host computer is designated as the master and that further away the slave.

Since there are six possible directions in which data can be transmitted from a unit to another unit, it would be possible to provide six separate registers in the communication circuit for transfer in the six possible each directions. However, it is more economical to provide a single communication register and to use a multiplexer and a demultiplexer to connect the end terminals of the register selectively to the six possible other units. Another multiplexer handles the clock signal. Figure 8 shows how this could be done.

Should a data processing unit be non-functional or busy, no message can be passed to it, but will remain in the transmitting communication register because an exchange of ready signals preliminary to the transfer will not have taken place. If this occurs, the communication circuit is arranged to reroute the message via other units. In this way data can be arranged to bypass non-functional units or units which are busy. This recovery procedure may be defined purely in the software of the microprocessors and can be arranged to

implement whatever algorithms are thought best to suit particular circumstances.

As mentioned above, a broadcast channel is provided from the host computer to all of the units. This is intended for the transfer of messages from the host computer to all or selected ones of the units and may carry, for example, control messages such as 'start' and 'stop'. Each message includes an address part and a data part, the parts being preceded by control bits indicating whether it is followed by the address part or the data part. Each unit has a key register to which a message address is transferred from the message received from the host. If the message address received on the broadcast channel agrees with that to which the key register responds then the data is received by the unit and an interrupt sent to the microprocessor to enter the data or implement instructions as required. The matching of the message addresses with the key may be restricted to bits of a single type, for example, 1. If the message address has 1's in all places where the key register has 1's, then the unit receives the message, and the presence of further 1's in the message address is ignored. This enables a global message address consisting entirely of 1's to be sent to every unit whilst retaining the flexibility of selectively addressed messages when required.

Figure 9 illustrates the functions executed by the gate array 4 which acts as the solution engine. The solution engine evaluates a neuron by deriving from the dynamic RAM 2 the terms of the input vector in turn and multiplies them by the corresponding synapse values also read from the DRAM 2. The addresses of the DRAM 2 are generated by a functional unit 40 and the reading or writing instruction by a unit 41. Data control unit 42 includes registers for storing the data read from the DRAM 2. The unit 42 also applies the accumulated products of the terms of the input vector and the corresponding synapse values produced by multiplier 43 and summed by adder 44 and stored in a register 45 back to an input of the adder 44. When a neuron has been fully solved its value appears in the register 45 and is transferred by a

second data control unit 46 to the microprocessor 1 (Figure 2) for application of the non-linear function to the value and the derivation of the term of the output vector of the data processing unit as described above.

The solution engine also includes an instruction decoder 47 which causes the engine to execute the predetermined sequence of operations outlined above for producing the neuron values. It also enables the input vectors and the original and updated synapse values to be stored in a planned series of addresses in the DRAM 2.

CLAIMS:

1. A computer comprising a plurality of separate data processing units each including a processor, a multi-address read-write memory and communication means, wherein the data processing units are arranged in a three-dimensional, possibly rectangular, array and the communication means of each unit is connected to the communication means of the nearest units only in each direction along the three axes of the array.

2. A computer comprising a plurality of separate data processing units each including a processor, a multi-address read-write memory and communication means, the communication means of each unit being connected to the communication means of at least two other data processing units or to the communication means of at least one other data processing unit and to a data input or a data output, wherein each data processing unit operates as a neuron simulator responding to a plurality of data bits as an input and producing a plurality of data bits as an output.

3. A computer according to claim 2 wherein each processor includes a gate array configured as a neural network solution engine capable of multiplying a plurality of input numbers by respective coefficients, summing the products to produce a single number and performing a non-linear operation on the single number to produce an output number.

4. A computer according to claim 2 or 3, wherein each data processing unit includes a microprocessor to control the unit and determine to which other unit communication is to be effected.

5. A computer according to any preceding claim wherein the communication means of each data processing unit includes a shifting register which is arranged to provide parallel data transfer to and from other components of the particular data processing unit, and is connected to provide serial data transfer to and from the shifting register of the communication means of the or each other data processing unit to which it is connected.

6. A computer according to claim 5 wherein the shifting registers of the communication means of two data processing units which are connected together are connected in a loop so that while data is being transferred from one register to the other they are also being transferred from the other registers to the one.

7. A computer according to claim 6 wherein in the communication means of each data processing unit there are provided a single shifting register and two multiplexers for selectively connecting the serial input and output of the shifting register to the outputs and inputs of the shifting registers of the communication means of other data processing units.

8. A computer according to claim 7 wherein the communication means of each data processing unit further includes a third multiplexer for linking the clock of the communication means to the clock of another communication means so as to synchronise the stepping of the data of the two shifting registers connected together in a loop.

9. A computer according to any one of claims 2 to 8 wherein each data processing unit has six pairs of terminals for connecting the communication means of the particular data processing unit to the communication means of six other data processing units.

10. A computer according to claim 9 wherein the data processing units are arranged in a three-dimensional rectangular array and each unit is connected to communicate with the nearest units in the three coordinate directions.

11. A computer according to any one of claims 2 to 10 further including broadcasting means connected to a receive-only input of each data processing unit.

12. A computer according to claim 11 wherein each data processing unit has allocated to it a multi-bit address and includes message selection means connected to the receive-only input for accepting only those messages from the broadcasting means which are addressed to the particular processing unit.

13. A computer substantially as described herein and as illustrated by the accompanying drawings.

BNSDOCID: <GB 2206428A>